

# 2017년 프로그래밍언어 개론

## -homework 01-

제출일자	2017.03.07.
이름	정 윤 수
학 번	201302482
분 반	03

## 1. sum(n)

```
def sum(n):  
    if n==0:  
        return 0  
    return n + sum(n-1)
```

### 문제해결 방식

- 이 문제는 1부터 n까지의 합을 재귀를 이용 하여 구하는 함수이다. 먼저 Recursion을 끝낼 수 있는 base 사례부터 생각을 해보았다. n의 값을 하나씩 줄여가면서 n의 값이 0이 되었을 때 값을 0을 반환을 하면 재귀 함수를 빠져 나올 수 있을 것이라 생각을 하였고 나머지 숫자들을 덧셈을 해주면 반복문을 사용을 하지 않고 n까지의 합을 구할 수 있을것이라고 생각을 하였다.

## 2. fibonacci(n)

```
def fibonacci(n):  
    if n==1:  
        return 1  
    elif n==2:  
        return 1  
    return fibonacci(n-1)+fibonacci(n-2)
```

### 문제해결 방식

- 이 함수는 피보나치 수열에서 n번째의 수를 반환을 해주는 함수이다. 피보나치 수열은 첫 번째 수와 두 번째 수의 값이 1이다. 나는 이것을 base사례로 이용을 하였다. 세 번째 피보나치 수열의 수는 첫 번째와 두 번째 수를 이용을 하여 구할 수 있다. 네 번째 수는 두 번째 수와 세 번째 수를 이용을 하여 구할 수 있지만 세 번째 수는 첫 번째와 두 번째 수를 이용을 하여 구한다. 이러한 방식으로 생각을 하여 피보나치 수열의 n번째 값도 이런 식으로 구할 수 있을것이라 생각을 하였다.

### 3. factorial(n)

```
def factorial(n):  
    if n==0:  
        return 1  
    return n * factorial(n-1)
```

#### 문제해결 방식

- factorial(n) 함수는 1부터 n까지의 수의 곱을 구하는 함수이다. 이 factorial을 구하는 함수는 sum을 구하는 함수와 비슷한 원리로 동작을 할 것이라고 생각했다. 그 이유는 서로 1부터 n까지의 수를 이용을 한다는 방식에서 생각을 하였다. base사례로 n의 값이 0이면 1을 반환을 해주고 나머지 값들과 곱셈을 해주면서 n까지의 곱을 재귀를 이용을 하여 쉽게 계산을 할 수 있을 것이라고 생각했다.

### 4. demical\_to\_binary(n)

```
def decimal_to_binary(n):  
    if n <= 1:  
        return n  
    return n % 2 + 10 * decimal_to_binary(n/2)
```

#### 문제해결 방식

- decimal\_to\_binary(n) 함수는 10진수를 2진수로 변경을 하는 함수이다. 일단 맨 처음 함수를 작성하기 전 먼저 base사례를 생각을 해 보았다. 10진수를 2진수로 바꾸기 위해서는 1이 남을 때 까지 계속 나누기 2를 해야 한다. 그렇게 1이 남게 되면 그 값이 이진수의 가장 앞의 값이 된다. 나머지 자리들은 각 나눗셈 연산에서의 나머지 값들을 이용을 하여 채워주면 된다. 이런 식으로 생각을 하게 되어 n의 값이 1 일 때 1을 반환을 해주면 이진수 가장 앞 자리를 만들 수 있고 나머지 자리들은 그 연산에 나머지 값들을 채워넣어 주면 될 것이라고 생각을 하였다. 또한 2진수 자릿수를 구현을 하기위해서 재귀를 할 때마다 10을 곱해 주었다 그렇게 하면 2진수의 자릿수를 구현을 할 수 있다.

#### 결과

```
C:\Python27\python.exe C:/Users/정윤수/PycharmProjects/201302482_정윤수_01/Recusion.py  
5050  
55  
120  
1111
```

## 느낀점

- 파이썬을 처음 사용을 해보면서 java와 c언어와는 다르게 어떤 식으로 프로그래밍을 할수 있을까? 또한 내가 이것으로 과제들을 해결 할수있을까? 라고 맨 처음에 생각이 들었다. 하지만 공부를 해보면서 파이썬이 매우 단순하고 다른 언어들을 먼저 선행을 한 후 하면 배우기 매우 쉽다고 생각이 들었고. 자바와 c언어에서의 복잡한 괄호들을 볼 필요가 거의 없다는 것에서 괄호 실수를 하지 않을 것 이라는 생각이 들어서 마음이 편하였다 또한 사용을 해보면서 매우 간편하다고 생각하였다. 자바와 c언어로 구현을 하였던 재귀 함수들을 파이썬으로 구현을 해보니 어렵지 않게 구현을 할수 있었다.